

GUI Components: Part I, Solutions

14

*Do you think I can listen all day
to such stuff?*

—Lewis Carroll

*Even a minor event in the life of
a child is an event of that child's
world and thus a world event.*

—Gaston Bachelard

*You pays your money and you
takes your choice.*

—Punch

Objectives

In this chapter you'll learn:

- Design principles of graphical user interfaces (GUIs).
- How to use Java's elegant, cross-platform Nimbus look-and-feel.
- To build GUIs and handle events generated by user interactions with GUIs.
- To understand the packages containing GUI components, event-handling classes and interfaces.
- To create and manipulate buttons, labels, lists, text fields and panels.
- To handle mouse events and keyboard events.
- To use layout managers to arrange GUI components

Self-Review Exercises

14.1 Fill in the blanks in each of the following statements:

- a) Method _____ is called when the mouse is moved with no buttons pressed and an event listener is registered to handle the event.

ANS: `mouseMoved`.

- b) Text that cannot be modified by the user is called _____ text.

ANS: `uneditable` (read-only).

- c) A(n) _____ arranges GUI components in a Container.

ANS: layout manager.

- d) The `add` method for attaching GUI components is a method of class _____.

ANS: `Container`.

- e) GUI is an acronym for _____.

ANS: graphical user interface.

- f) Method _____ is used to specify the layout manager for a container.

ANS: `setLayout`.

- g) A `mouseDragged` method call is preceded by a(n) _____ method call and followed by a(n) _____ method call.

ANS: `mousePressed`, `mouseReleased`.

- h) Class _____ contains methods that display message dialogs and input dialogs.

ANS: `JOptionPane`.

- i) An input dialog capable of receiving input from the user is displayed with method _____ of class _____.

ANS: `showInputDialog`, `JOptionPane`.

- j) A dialog capable of displaying a message to the user is displayed with method _____ of class _____.

ANS: `showMessageDialog`, `JOptionPane`.

- k) Both `JTextField`s and `JTextArea`s directly extend class _____.

ANS: `JTextComponent`.

14.2 Determine whether each statement is *true* or *false*. If *false*, explain why.

- a) `BorderLayout` is the default layout manager for a `JFrame`'s content pane.

ANS: True.

- b) When the mouse cursor is moved into the bounds of a GUI component, method `mouseover` is called.

ANS: False. Method `mouseenter` is called.

- c) A `JPanel` cannot be added to another `JPanel`.

ANS: False. A `JPanel` can be added to another `JPanel`, because `JPanel` is an indirect subclass of `Component`. So, a `JPanel` is a `Component`. Any `Component` can be added to a `Container`.

- d) In a `BorderLayout`, two buttons added to the `NORTH` region will be placed side by side.

ANS: False. Only the last button added will be displayed. Remember that only one component should be added to each region in a `BorderLayout`.

- e) A maximum of five components can be added to a `BorderLayout`.

ANS: True. [Note: Panels containing multiple components can be added to each region.]

- f) Inner classes are not allowed to access the members of the enclosing class.

ANS: False. Inner classes have access to all members of the enclosing class declaration.

- g) A `JTextArea`'s text is always read-only.

ANS: False. `JTextAreas` are editable by default.

- h) Class `JTextArea` is a direct subclass of class `Component`.

ANS: False. `JTextArea` derives from class `JTextComponent`.

- 14.3** Find the error(s) in each of the following statements, and explain how to correct it (them):
- a) `buttonName = JButton("Caption");`
 ANS: `new` is needed to create an object.
- b) `JLabel aLabel, JLabel; // create references`
 ANS: `JLabel` is a class name and cannot be used as a variable name.
- c) `textField = new JTextField(50, "Default Text");`
 ANS: The arguments passed to the constructor are reversed. The `String` must be passed first.
- d) `setLayout(new BorderLayout());`
`button1 = new JButton("North Star");`
`button2 = new JButton("South Pole");`
`add(button1);`
`add(button2);`
 ANS: `BorderLayout` has been set, and components are being added without specifying the region, so both are added to the center region. Proper `add` statements might be
`add(button1, BorderLayout.NORTH);`
`add(button2, BorderLayout.SOUTH);`

Exercises

NOTE: Solutions to the programming exercises are located in the `ch14solutions` folder. Each exercise has its own folder named `ex14_##` where `##` is a two-digit number representing the exercise number. For example, exercise 14.12's solution is located in the folder `ex14_12`. At the end of this PDF we list compilation issues for exercise code compiled with Java SE 7.

- 14.4** Fill in the blanks in each of the following statements:
- a) The `JTextField` class directly extends class _____.
 ANS: `JTextComponent`.
- b) Container method _____ attaches a GUI component to a container.
 ANS: `add`.
- c) Method _____ is called when a mouse button is released (without moving the mouse).
 ANS: `mouseClicked`.
- d) The _____ class is used to create a group of `JRadioButtons`.
 ANS: `ButtonGroup`.
- 14.5** Determine whether each statement is *true* or *false*. If *false*, explain why.
- a) Only one layout manager can be used per `Container`.
 ANS: `True`.
- b) GUI components can be added to a `Container` in any order in a `BorderLayout`.
 ANS: `True`.
- c) `JRadioButtons` provide a series of mutually exclusive options (i.e., only one can be true at a time).
 ANS: `True`.
- d) `Graphics` method `setFont` is used to set the font for text fields.
 ANS: `False`. Component method `setFont` is used.
- e) A `JList` displays a scrollbar if there are more items in the list than can be displayed.
 ANS: `False`. A `JList` never provides a scrollbar.
- f) A `Mouse` object has a method called `mouseDragged`.
 ANS: `False`. A `Mouse` object is not provided by Java.

4 Chapter 14 GUI Components: Part I, Solutions

14.6 Determine whether each statement is *true* or *false*. If *false*, explain why.

a) A `JPanel` is a `JComponent`.

ANS: True.

b) A `JPanel` is a `Component`.

ANS: True.

c) A `JLabel` is a `Container`.

ANS: True.

d) A `JList` is a `JPanel`.

ANS: False. A `JList` is a `JComponent`.

e) An `AbstractButton` is a `JButton`.

ANS: False. A `JButton` is an `AbstractButton`.

f) A `TextField` is an `Object`.

ANS: True.

g) `ButtonGroup` is a subclass of `JComponent`.

ANS: False. `ButtonGroup` inherits from `Object`.

14.7 Find any errors in each of the following lines of code, and explain how to correct them.

a) `import javax.swing.JFrame`

ANS: Semicolon is missing after the class name.

b) `pane1Object.setLayout(8, 8); // set GridLayout`

ANS: The `GridLayout` constructor cannot be used in this manner. The correct statement should be:

```
pane1Object.getContentPane().setLayout( new GridLayout( 8, 8 ) );
```

c) `container.setLayout(new FlowLayout(FlowLayout.DEFAULT));`

ANS: Class `FlowLayout` does not contain static constant `DEFAULT`.

d) `container.add(eastButton, EAST); // BorderLayout`

ANS: `EAST` should be `BorderLayout.EAST`.

Java SE 7 Compilation Issues

14.15 This exercise compiles with warnings, because, as of Java SE 7, `JList` is a generic class. For the 10th edition, we'll update this solution to use the new version of `JList`.